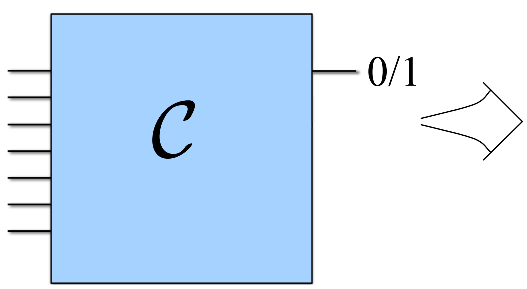
Techniques for fault-tolerant quantum error correction

Ben Reichardt
UC Berkeley

Quantum fault-tolerance problem

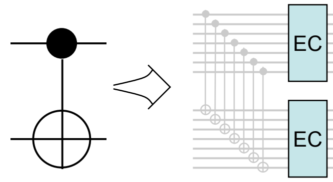


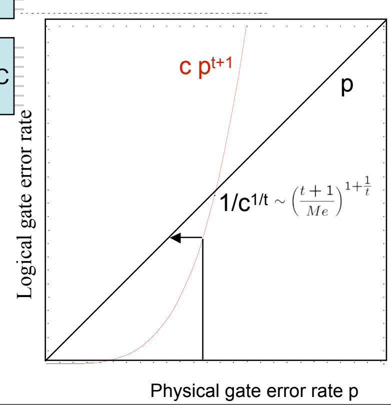
Fault-tolerant, larger
$\mathcal{C}$

- High tolerable noise
- Low overhead

Encoding for fault tolerance

- **Idea:** Encode ideal/logical circuit into quantum error-correcting code. Apply gates directly on the encoded data, each gate followed by error correction.
  - m-qubit, t-error correcting code
    [[m, 1, d=2t+1]]





\# CNOT locations

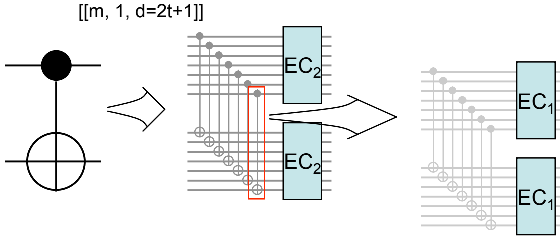
$$C_1 \leq \binom{M}{t+1} C_0^{t+1}$$

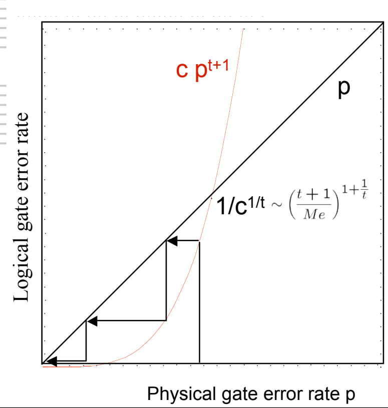
level-1 CNOT failure rate

physical failure rate

Concatenated encoding for arbitrary accuracy

- **Idea:** Encode ideal/logical circuit into quantum error-correcting code. Apply gates directly on the encoded data, each gate followed by error correction.
  - m-qubit, t-error correcting code
    [[m, 1, d=2t+1]]





\# CNOT locations

$$C_k \leq \binom{M}{t+1} (C_{k-1})^{t+1}$$

level-k CNOT failure rate

level-(k-1) failure rate

Threshold theorems

For a physical error rate $\varepsilon < \varepsilon_c$, an N-gate ideal quantum circuit can be reliably simulated with N poly(log N) physical gates.

Examples:

- Independent probabilistic noise
  - $\varepsilon_c > 0$ [Aharonov & Ben-Or '97, Kitaev '97]
  - $\varepsilon_c > 2.7 \times 10^{-5}$ [Aliferis, Gottesman, Preskill '05]
  - $\varepsilon_c > 6 \times 10^{-6}$ with Pauli errors [R '05]
  - $\varepsilon_c \geq 10^{-4}$ (today)
  - $\varepsilon_c = 1/2$ for Bell measurement erasure errors (detected errors) [Knill '03]

Fault-tolerance threshold myths:

Independent probabilistic noise.
Nonlocal gates.
Maximize the threshold regardless of the overhead.

Threshold theorems

For a physical error rate $\varepsilon < \varepsilon_c$, an N-gate ideal quantum circuit can be reliably simulated with N poly(log N) physical gates.

Examples:

- Independent probabilistic noise
  - $\varepsilon_c > 0$ [Aharonov & Ben-Or '97, Kitaev '97]
  - $\varepsilon_c > 2.7 \times 10^{-5}$ [Aliferis, Gottesman, Preskill '05]
  - $\varepsilon_c > 6 \times 10^{-6}$ with Pauli errors [R '05]
  - $\varepsilon_c \geq 10^{-4}$ (today)
  - $\varepsilon_c = 1/2$ for Bell measurement erasure errors (detected errors) [Knill '03]

- Non-Markovian local noise [Terhal/Burkard '04, Aliferis/Gottesman/Preskill '05]
- Correlated noise [Knill/Laflamme/Zurek '97]
- Local interactions
  - 2D grid (nearest n'bor), 1D line (next-nearest) [Gottesman '99]
  - with correlated noise [Aharonov, Kitaev, Preskill '05]

Outline

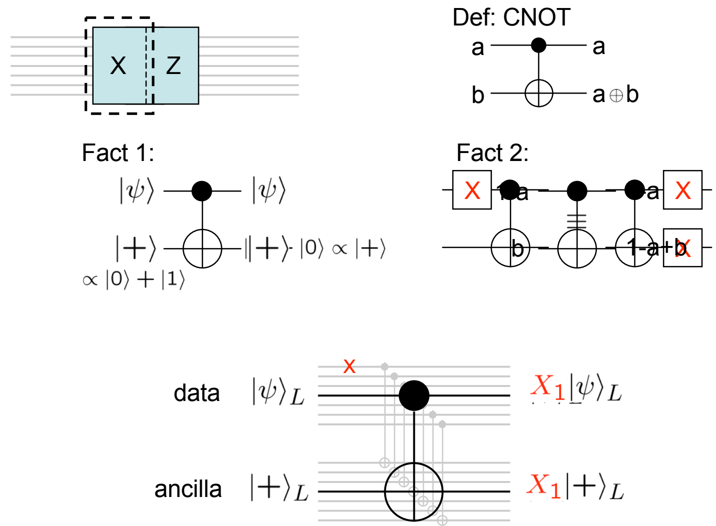
- **Idea** for improved ancilla verification for error correction: Differently prepare ancillas to verify against each other
  - Makes postselection unnecessary with 7-qubit Steane code [Aliferis]
  - Halves preparation complexity for 23-qubit Golay code (1200 → 600 CNOT gates). Allows detailed combinatorial analysis to show high provable threshold (10^{-4})

Outline

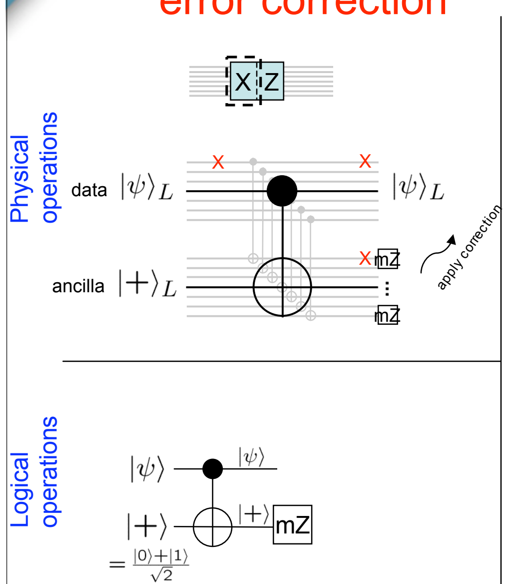
- **Idea:** Differently prepare ancillas to verify against each other
  - No postselection for Steane code [Aliferis]
  - Halves preparation complexity for 23-qubit Golay code

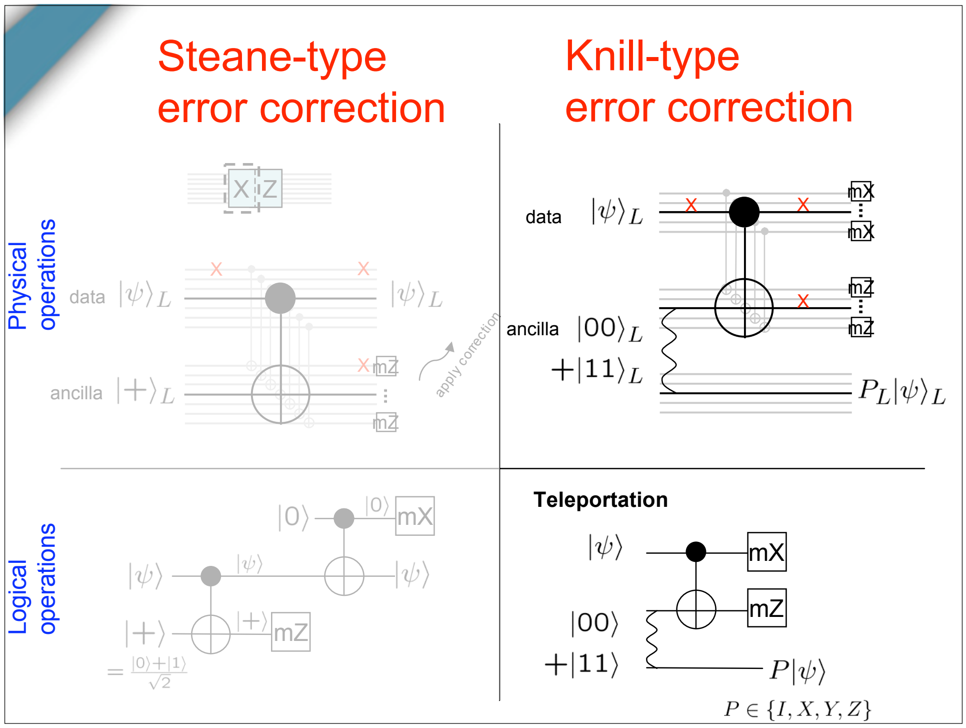
- Technical background
  - Error correction
  - Quantum ECCs
  - Stabilizer algebra
- Ancilla preparation and verification
  - Steane preparation and heuristic verification
    - for Steane 7-qubit, distance-3 code
    - for Bacon/Shor 9-qubit, distance-3 code
  - Strictly fault-tolerant verification
    - repeated purification
    - tweaked
- Rigorous noise threshold for 23-qubit, distance-7 Golay code
  - Technical setup
  - Combinatorial analysis

Steane-type error correction

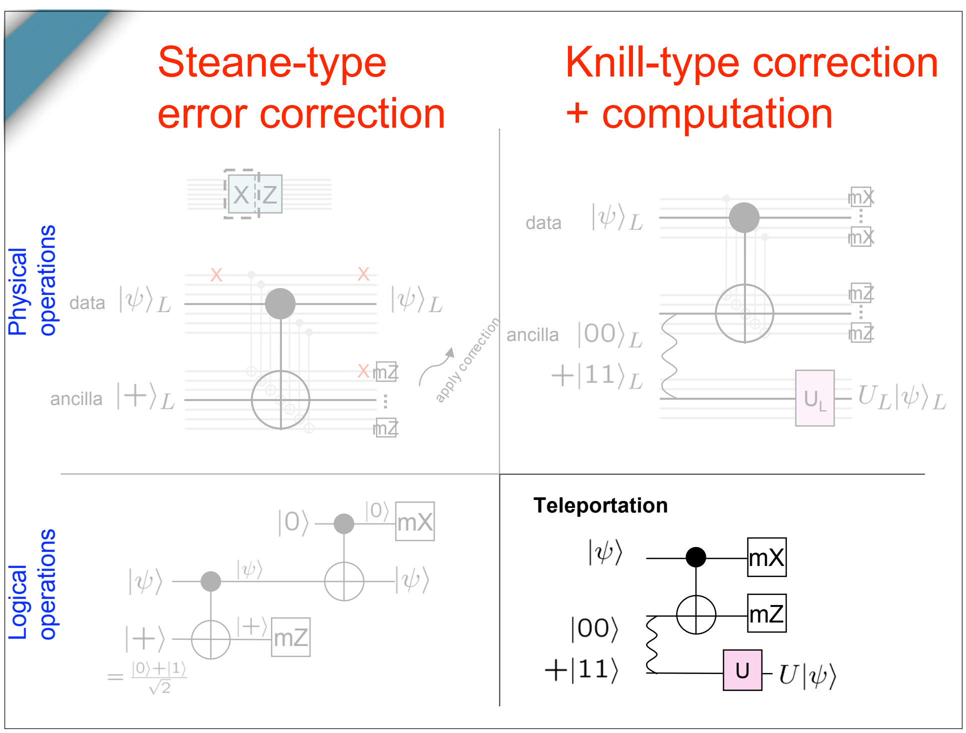


Steane-type error correction

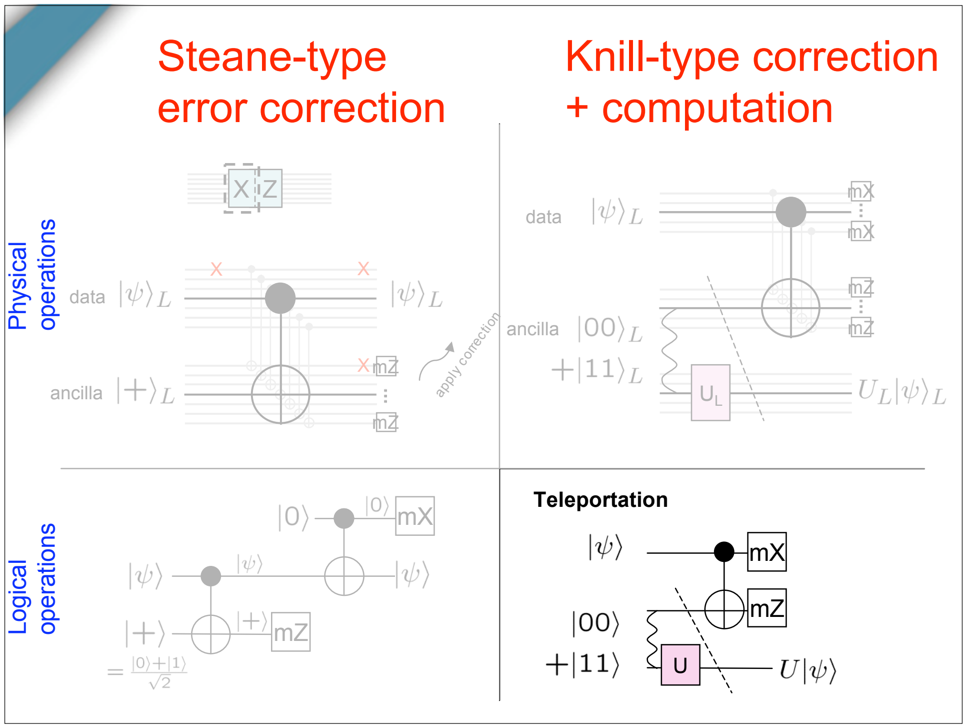




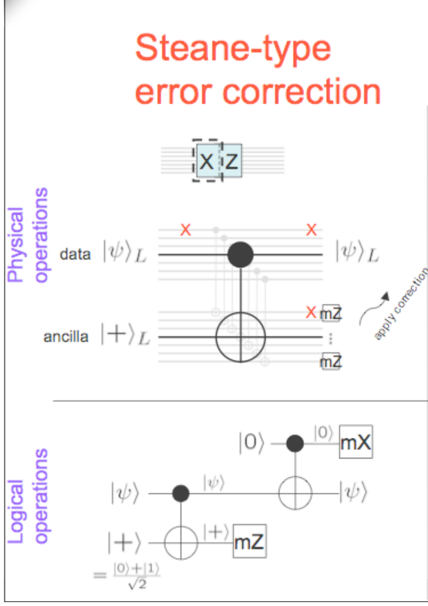
Steane-type
error correction
Knill-type
error correction
Physical operations
Logical operations
data
ancilla
apply correction
mX
mZ
Teleportation
$|\psi\rangle_L$
$|+\rangle_L$
$|00\rangle_L + |11\rangle_L$
$P_L|\psi\rangle_L$
$|00\rangle + |11\rangle$
$P|\psi\rangle$
$= \frac{|0\rangle+|1\rangle}{\sqrt{2}}$
$P \in \{I, X, Y, Z\}$



Steane-type
error correction
Knill-type correction
+ computation
Physical operations
Logical operations
data
ancilla
apply correction
mX
mZ
Teleportation
U_L
U
$U_L|\psi\rangle_L$
$|00\rangle_L + |11\rangle_L$
$|00\rangle + |11\rangle$
$U|\psi\rangle$
$= \frac{|0\rangle+|1\rangle}{\sqrt{2}}$



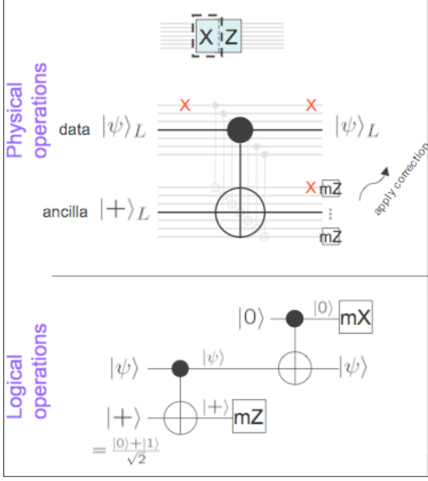
Error correction properties



- Arbitrary state is brought back into codespace (except with controlled errors: weight-k errors with probability $O(p^k)$)
- On states with controlled errors, no logical effect is applied (and errors remain controlled)

Remarks

Steane-type error correction



- Computation can "typically" continue without waiting for error-correction measurements to complete
  - (when correction information becomes available, propagate corrections through the circuit)
- High-fidelity ancillas do not suffice (need both high fidelity *and* uncorrelated errs)

$\Rightarrow$ Ancilla verification
  - Ancillas can't be used until verified, so computation has to wait for verification measurements to complete

$\Rightarrow$ Ancilla factories
  - Prepare many ancillas in parallel and in advance, so a verified ancilla is always ready

$\Rightarrow$ High overhead

Quantum error-correcting codes

$\mathcal{H} = A \oplus B$

codespace = simultaneous +1 eigenspace of code stabilizers

physical bits logical bits distance

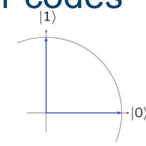
- [[n=4,k=2,d=2]] erasure code
  - used in Knill's fault-tolerance scheme together with certain [[6,2,2]] code
- [[5,1,3]] code
  - not CSS — stabilizer includes, e.g., XZZXI

<u>CSS code</u>: All stabilizers can be written as product of Xs or a product of Zs

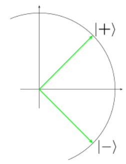
- Steane [[7,1,3]] code
- Bacon/Shor [[9,1,3]] operator ECC
- [[15,1,3]] Reed-Muller code
  - allows for transverse (X+Z)/√2 application (for universality), but not self-dual
- Golay [[23,1,7]] code

CSS quantum stabilizer codes

- Classical codewords in the 0/1 basis
  $\Rightarrow$ Correct bit flip X errors



- Classical codewords in the +/- basis
  $\Rightarrow$ Correct phase flip Z errors



- E.g., Steane [[7,1,3]] code corrects arbitrary error on one qubit
  - Based on classical Hamming [7,4,3] code

$$C^{\perp} = \begin{pmatrix} 0&0&0&1&1&1&1 \\ 0&1&1&0&0&1&1 \\ 1&0&1&0&1&0&1 \end{pmatrix}$$

$$C = \begin{pmatrix} 0&0&0&1&1&1&1 \\ 0&1&1&0&0&1&1 \\ 1&0&1&0&1&0&1 \\ 1&1&1&1&1&1&1 \end{pmatrix}$$

$$\begin{pmatrix} I&I&I&Z&Z&Z&Z \\ I&Z&Z&I&I&Z&Z \\ Z&I&Z&I&Z&I&Z \\ I&I&I&X&X&X&X \\ I&X&X&I&I&X&X \\ X&I&X&I&X&I&X \end{pmatrix}$$

$$\begin{aligned} X_L &= X^{\otimes 7} \\ Z_L &= Z^{\otimes 7} \end{aligned}$$

Steane [[7,1,3]] quantum code

- Corrects arbitrary error on one qubit
  - Based on classical Hamming [7,4,3] code
- Simultaneous +1 eigenspace of 6 independent Pauli “stabilizer” elements

$$\begin{pmatrix} I&I&I&Z&Z&Z&Z \\ I&Z&Z&I&I&Z&Z \\ Z&I&Z&I&Z&I&Z \\ I&I&I&X&X&X&X \\ I&X&X&I&I&X&X \\ X&I&X&I&X&I&X \end{pmatrix}$$

$$\mathcal{H} = A \oplus B$$

$$\begin{aligned} X_L &= X^{\otimes 7} \\ Z_L &= Z^{\otimes 7} \end{aligned}$$

$$S = \left\{ \begin{array}{r} 0^7, 0001111, 0110011, 0111100, \\ 1010101, 1011010, 1100110, 1101001 \end{array} \right\}$$

$$H_L = H^{\otimes 7}$$
$$CNOT_L = CNOT^{\otimes 7}$$

$$|0_L\rangle = \frac{1}{\sqrt{8}} \sum_{x \in S} |x\rangle \qquad |1_L\rangle = X^{\otimes 7} |0_L\rangle$$

Stabilizer algebra

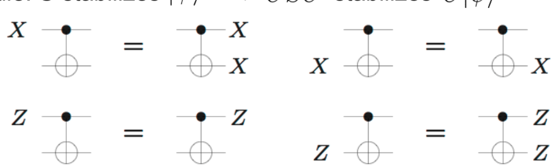
- Def: S *stabilizes* $|\psi\rangle$ if $S|\psi\rangle = |\psi\rangle$
- Rules:
  - S, T stabilize $|\psi\rangle \Rightarrow$ ST stabilizes $|\psi\rangle$
  - S stabilizes $|\psi\rangle \Rightarrow USU^\dagger$ stabilizes $U|\psi\rangle$
- Def: Pauli group = tensor products of Pauli operators I, X, Y or Z (with phase ±1 or ±i)
  - note all Paulis have half eigenvalues +1, half -1; pairs of Paulis either commute or anticommute
- Def: Stabilizer state on n qubits = intersection of +1 eigenspaces of n independent commuting Paulis
- Example:

Operation	State	Stabilizer $S = \{M \in \mathcal{P} : M|\psi\rangle = |\psi\rangle\}$
1. prepare $|+\rangle$	$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$	$\langle X \rangle$
2. prepare $|1\rangle$	$\frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)$	$\langle X \otimes I, I \otimes -Z \rangle$
3. CNOT$_{1,2}$	$\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$	$\langle XX, -ZZ \rangle$

$$\begin{array}{l} X \otimes I \to X \otimes X \\ Z \otimes I \to Z \otimes I \\ I \otimes X \to I \otimes X \\ I \otimes Z \to Z \otimes Z \end{array}$$

Stabilizer algebra

- Rule: S stabilizes $|\psi\rangle \Rightarrow USU^\dagger$ stabilizes $U|\psi\rangle$



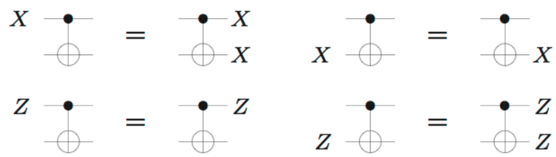
- Def: Stabilizer state on n qubits = intersection of +1 eigenspaces of n independent commuting Paulis
- Example:

Operation	State	Stabilizer $S = \{M \in \mathcal{P} : M|\psi\rangle = |\psi\rangle\}$
1. prepare $|+\rangle$	$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$	$\langle X \rangle$
2. prepare $|1\rangle$	$\frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)$	$\langle X \otimes I, I \otimes -Z \rangle$
3. CNOT$_{1,2}$	$\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$	$\langle XX, -ZZ \rangle$

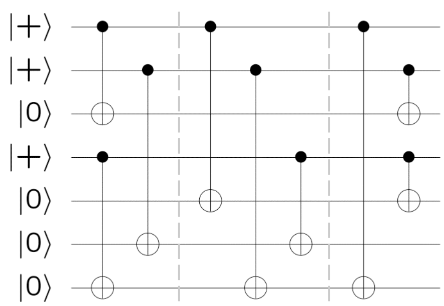
$$\begin{array}{l} X \otimes I \to X \otimes X \\ Z \otimes I \to Z \otimes I \\ I \otimes X \to I \otimes X \\ I \otimes Z \to Z \otimes Z \end{array}$$

Stabilizer algebra

- Rule: S stabilizes $|\psi\rangle$ $\Rightarrow USU^\dagger$ stabilizes $U|\psi\rangle$



- Example:



Initial stabilizers		Final stabilizers
XIIIIII	→	XIXIXIX
IXIIIII	→	IXXIIXX
IIIXIII	→	IIIXXXX
IIZIIII	→	ZZZIZIZ
IIIIZII	→	ZZZZZZZ
IIIIIZI	→	IZIZZZZ
IIIIIIZ	→	ZZZZZZZ

Steane code $|0\rangle_L$

Outline

- **Idea:** Differently prepare ancillas to verify against each other
  - No postselection for Steane code [Aliferis]
  - Halves preparation complexity for 23-qubit Golay code

- Technical background
  - Error correction
  - Quantum ECCs
  - Stabilizer algebra
- Ancilla preparation and verification
  - Steane preparation and heuristic verification
    - for Steane 7-qubit, distance-3 code
    - for Bacon/Shor 9-qubit, distance-3 code
    - for higher-distance codes
  - Strictly fault-tolerant verification
    - repeated purification
    - tweaked
- Rigorous noise threshold for 23-qubit, distance-7 Golay code
  - Technical setup
  - Combinatorial analysis

Steane encoded ancilla preparation

1. Using Gaussian elimination, and by rearranging qubits, put state's X (or Z) generators in standard form.

$k\{\ \overbrace{I}^{k}\,|\,\overbrace{A}^{n-k}$

(or $A^T | I$)

e.g.

$$\begin{array}{ccccccc} 1 & 1 & 1 & X & X & X & X \\ 1 & X & X & 1 & 1 & X & X \\ X & 1 & X & 1 & X & 1 & X \\ \uparrow & \uparrow & & \uparrow & & & \end{array}$$

$$\Rightarrow \begin{array}{ccc|cccc} X & \cdot & \cdot & X & X & \cdot & X \\ \cdot & X & \cdot & X & \cdot & X & X \\ \cdot & \cdot & X & \cdot & X & X & X \end{array}$$

$\underbrace{\qquad\qquad}_{A}$

Steane encoded ancilla preparation

1. Using Gaussian elimination, and by rearranging qubits, put state's X (or Z) generators in standard form.

$k\{\ \overbrace{I}^{k}\,|\,\overbrace{A}^{n-k}$

(or $A^T | I$)

e.g.

$$\begin{array}{ccccccc} 1 & 1 & 1 & X & X & X & X \\ 1 & X & X & 1 & 1 & X & X \\ X & 1 & X & 1 & X & 1 & X \\ \uparrow & \uparrow & & \uparrow & & & \end{array}$$

$$\Rightarrow \begin{array}{ccc|cccc} X & \cdot & \cdot & X & X & \cdot & X \\ \cdot & X & \cdot & X & \cdot & X & X \\ \cdot & \cdot & X & \cdot & X & X & X \end{array}$$

$\underbrace{\qquad\qquad}_{A}$

2. Starting with $|+^k 0^{n-k}\rangle$, use CNOT gates from first k qubits into last n-k qubits to generate each stabilizer.

eg.

control qubits ↓ target qubits ↓

initial X stabilizers:
$$\begin{array}{ccc|cccc} X & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & X & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & X & \cdot & \cdot & \cdot & \cdot \end{array}$$

${}^{c}_{1}X_4,\ {}^{c}_{1}X_5,\ {}^{c}_{1}X_7$

$$\Rightarrow \begin{array}{ccc|cccc} X & \cdot & \cdot & X & X & \cdot & X \\ \cdot & X & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & X & \cdot & \cdot & \cdot & \cdot \end{array}$$

${}^{c}_{2}X_4,\ {}^{c}_{2}X_6,\ {}^{c}_{2}X_7$

${}^{c}_{3}X_5,\ {}^{c}_{3}X_6,\ {}^{c}_{3}X_7$

Z stabilizers are correctly generated automatically.

Steane encoded ancilla preparation

1. Using Gaussian elimination, and by rearranging qubits, put state's X (or Z) generators in standard form.

$$k\{\ \overbrace{I}^{k}\,\big|\,\overbrace{A}^{n-k}$$

(or $A^T | I$)

e.g.

$$\begin{array}{ccccccc} I & I & I & X & X & X & X \\ I & X & X & I & I & X & X \\ X & I & X & I & X & I & X \\ \uparrow & \uparrow & & \uparrow & & & \end{array}$$

$$\Rightarrow \begin{array}{ccc|cccc} X & \cdot & \cdot & X & X & \cdot & X \\ \cdot & X & \cdot & X & \cdot & X & X \\ \cdot & \cdot & X & \cdot & X & X & X \end{array}$$

$\underbrace{\qquad}_{A}$

2. Starting with $|+^k 0^{n-k}\rangle$, use CNOT gates from first k qubits into last n−k qubits to generate each stabilizer.

e.g.

control qubits ↓ target qubits ↓

initial X stabilizers: $\begin{array}{ccc|cccc} X & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & X & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & X & \cdot & \cdot & \cdot & \cdot \end{array}$

${}^{c}_{1}X_4,\ {}^{c}_{1}X_5,\ {}^{c}_{1}X_7$

$$\Rightarrow \begin{array}{ccc|cccc} X & \cdot & \cdot & X & X & \cdot & X \\ \cdot & X & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & X & \cdot & \cdot & \cdot & \cdot \end{array}$$

${}^{c}_{2}X_4,\ {}^{c}_{2}X_6,\ {}^{c}_{2}X_7$

${}^{c}_{3}X_5,\ {}^{c}_{3}X_6,\ {}^{c}_{3}X_7$

Z stabilizers are correctly generated automatically.

3. Gates all commute, so rearrange them to maximize parallelism.
   a. In each time step, each control qubit can be used at most once.
   b. . . . And each target qubit can be targeted at most once.

Schedule corresponds to filling in nontrivial entries of A with round numbers.

$$\begin{array}{c|cccc} & 4 & 5 & 6 & 7 \\ \hline 1 & 1 & 2 & \cdot & 3 \\ 2 & 3 & \cdot & 1 & 2 \\ 3 & \cdot & 3 & 2 & 1 \end{array} \quad \longleftrightarrow \quad \begin{array}{l} \text{round 1: } {}^{c}_{1}X_4,\ {}^{c}_{2}X_6,\ {}^{c}_{3}X_7 \\ \text{round 2: } {}^{c}_{1}X_5,\ {}^{c}_{2}X_7,\ {}^{c}_{3}X_6 \\ \text{round 3: } {}^{c}_{1}X_7,\ {}^{c}_{2}X_4,\ {}^{c}_{3}X_5 \end{array}$$

a ⟺ no round # appears twice in a row

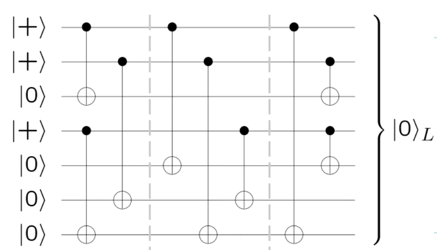
b ⟺ no round # appears twice in a column

⇝ # rounds ≥ max. no. nonzero entries in a row or column of A

"Latin rectangle" Hall's marriage theorem ⇒ equality suffices

Steane heuristic verification

- Steane $|0\rangle_L$ encoding circuit:



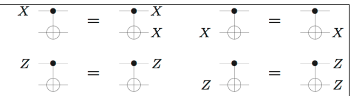
$$\begin{pmatrix} I & I & I & Z & Z & Z & Z \\ I & Z & Z & I & I & Z & Z \\ Z & I & Z & I & Z & I & Z \\ I & I & I & X & X & X & X \\ I & X & X & I & I & X & X \\ X & I & X & I & X & I & X \end{pmatrix}$$

$$X_L = X^{\otimes 7}$$
$$Z_L = Z^{\otimes 7}$$

- Gives correlated errors
  - e.g., weight-two X errors occur with 1st-order probability

  ⇒ Verification against X errors is required for fault tolerance
- Z errors are not correlated, so Z error verification is not required.
  - Z_L ~ ZZZ has no effect on $|0\rangle_L$; ⇒ two-bit error ZZI has same effect as IIZ, so all Z errors have reduced weight either 0 or 1.

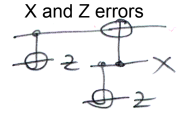
Steane heuristic verification



- Purification: Prepare two ancillas, check one against the other. Postselect on no detected errors in second ancilla.

X errors

X and Z errors



- In general: (but with a distance-3 code, this simplifies)

error weight	0	1	2	3	4	. . .
error order	0	1	2	2	2	. . .

- Steane finds, roughly, that one round of purification works well (according to simulations). However, this is not *strictly* fault-tolerant for codes of distance > 3.

Def: Fault-tolerant: Weight >1 errors are at most second-order events

Suffices for threshold existence

Def: *Strictly* fault-tolerant: Weight-k errors are at most kth-order events, $k \leq t+1=(d+1)/2$

Required for $p \rightarrow p^{t+1}$ effective error behavior

Encoding complexities

code type	$[[n,$ # qubits	$k,$ # encoded qubits	$d]]$ distance	w # rounds	N_A # gates	
None	1	1	1	–	–	
Hamming	7	1	3	3	12	
Golay	23	1	7	~~11~~ 7	77	→ efficient
,,	21	3	5	7	63	
BCH	31	11	5	15	122	
QR	47	1	11	15	281	
,,	45	3	9	15	255	
,,	43	5	7	15	229	
BCH	63	27	7	27	350	
,,	63	39	5	27	328	
QR	79	1	15	27	801	
,,	77	3	13	27	759	
,,	75	5	11	27	713	
QR	103	1	19	31	1265	
,,	101	3	17	31	1215	
,,	99	5	15	31	1165	
,,	97	7	13	31	1119	
BCH	127	29	15	47	1939	
,,	127	43	13	47	1802	

[Steane, quant-ph/0207119]

Encoding complexity can depend on code presentation.

Avoiding verification: Bacon/Shor 9-qubit code

- Shor's code: Concatenate 3-qubit repetition code with its dual
  - Repetition code: 0 → 000, 1 → 111

    Stabilizers ZZI, IZZ, ZIZ.
    Logical X is XXX, logical Z is ZII ~ IZI ~ IIZ.
    Corrects one bit flip (X) error.

  - Dual repetition code: $|+\rangle \to |+++\rangle, \quad |-\rangle \to |---\rangle$

    Stabilizers XXI, IXX, XIX.
    Logical Z is ZZZ, logical X is XII ~ IXI ~ IIX.
    Corrects one phase flip (Z) error.

  - Concatenation:

    Corrects one X error in each block of three, and one Z error.

Stabilizer generators:

$$
\begin{array}{r ccccccccc}
 & Z & Z & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & \cdot & Z & Z & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & \cdot & \cdot & \cdot & Z & Z & \cdot & \cdot & \cdot & \cdot \\
 & \cdot & \cdot & \cdot & \cdot & Z & Z & \cdot & \cdot & \cdot \\
 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & Z & Z & \cdot \\
 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & Z & Z \\
 & X & X & X & X & X & X & \cdot & \cdot & \cdot \\
 & \cdot & \cdot & \cdot & X & X & X & X & X & X \\
\hline
X_L = & X & X & X & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
Z_L = & Z & \cdot & \cdot & Z & \cdot & \cdot & Z & \cdot & \cdot
\end{array}
$$

- Bacon: Remove code redundancies
  - Operator error-correcting code $\mathcal{H} = (A \otimes B) \oplus C$

Ike covered this…

- Shor's code: Concatenate 3-qubit repetition code with its dual
- Preparing encoded ancilla $|+\rangle_L$:

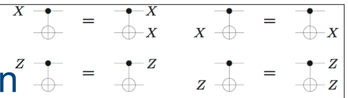
Stabilizer generators:

$$
\begin{array}{ccccccccc}
Z & Z & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & Z & Z & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & Z & Z & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & Z & Z & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & Z & Z & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & Z & Z \\
X & X & X & X & X & X & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & X & X & X & X & X & X \\
X & X & X & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}
\sim
\begin{array}{ccccccccc}
Z & Z & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & Z & Z & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & Z & Z & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & Z & Z & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & Z & Z & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & Z & Z \\
\cdot & \cdot & \cdot & X & X & X & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X & X \\
X & X & X & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}
$$

Thus $|+\rangle_L = (|000\rangle + |111\rangle)^{\otimes 3}$ and requires no Z verification. [Aliferis]

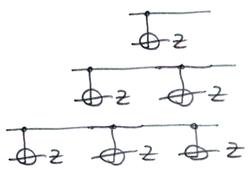
- Bacon: Restore X/Z symmetry

Golay code naïve verification



Def: Fault-tolerant: Weight >1 errors are at most second-order events
Def: *Strictly* fault-tolerant: Weight-k errors are at most kth-order events, k ≤ t+1=(d+1)/2

- Purification: Prepare two ancillas, check one against the other. Postselect on no detected errors in second ancilla.
- In general, repeated purification:



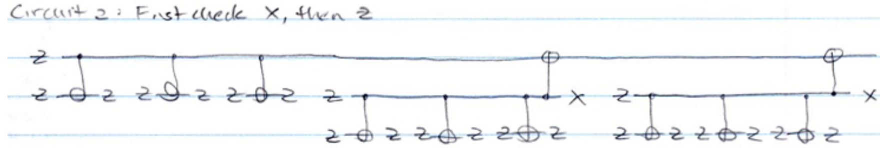
	X Error weight	0	1	2	3	4
Error order with	0 verifications	0	1	1	1	1
	1 verification	0	1	2	2	2
	2 verifications	0	1	2	3	3
	3 verifications	0	1	2	3	4

	Z Error weight	0	1	2	3
Error order with	0 verifications	0	1	1	1
	1 verification	0	1	2	2
	2 verifications	0	1	2	3

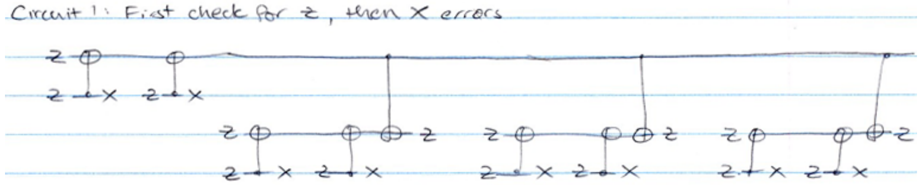
Golay code naïve verification

- For distance-seven code, generically need three rounds of verification against X errors, and two rounds of Z verification.
- Repeated purification circuits:

Circuit 2: First check X, then Z

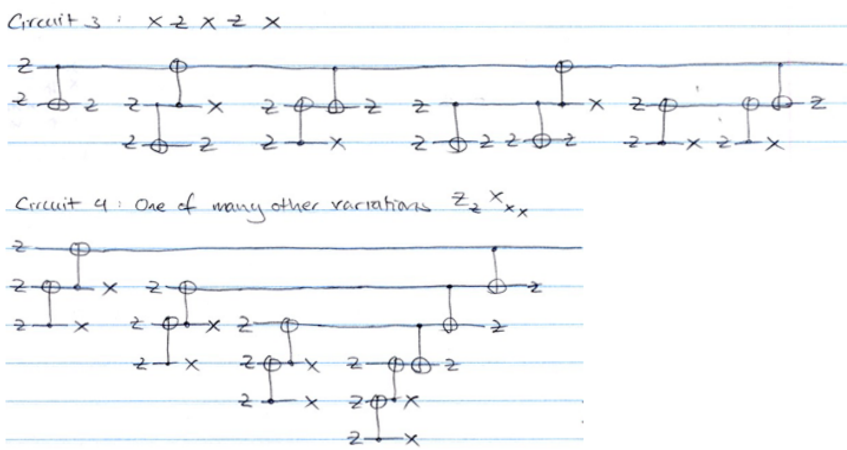


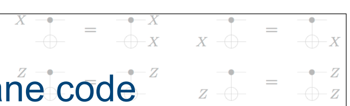
Circuit 1: First check for Z, then X errors



Golay code naïve verification

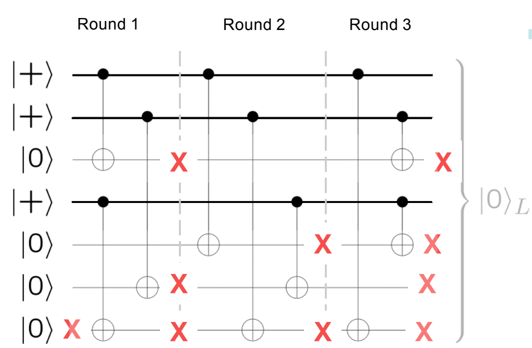
- Repeated purification circuits:





Smarter verification for Steane code

- Observe: X errors are correlated, but not arbitrary.

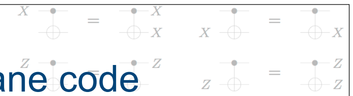


- Assume at most one X error occurs during preparation. What are the possible errors on the ouput?
  - Arbitrary single-bit errors (of course)
  - But what else?

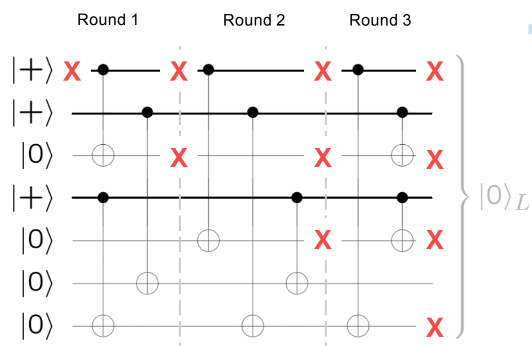
X stabilizers:

```
XIXIXIX
IXXIIXX
IIIXXXX
```

Smarter verification for Steane code



- Observe: X errors are correlated, but not arbitrary.



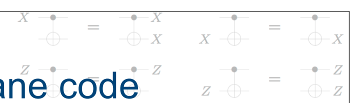
- Assume at most one X error occurs during preparation. What are the possible errors on the output?
  - Arbitrary single-bit errors (of course)
  - But what else?

~~$X_1 X_3 X_5 X_7$ ~I~~

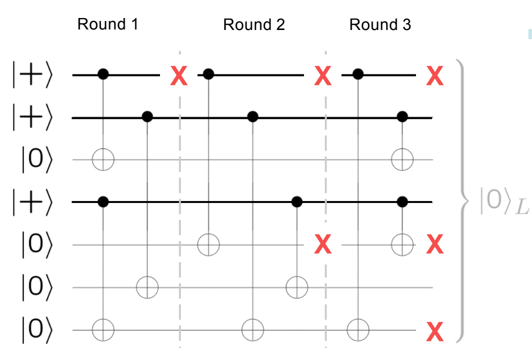
X stabilizers:

```
XIXIXIX
IXXIIXX
IIIXXXX
```

Smarter verification for Steane code



- Observe: X errors are correlated, but not arbitrary.



- Assume at most one X error occurs during preparation. What are the possible errors on the output?
  - Arbitrary single-bit errors (of course)
  - But what else?

~~$X_1 X_3 X_5 X_7$ ~I~~

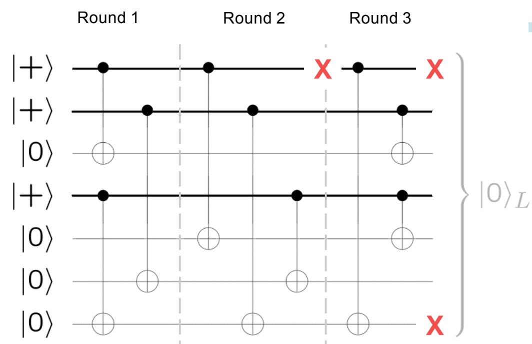
~~$X_1 X_5 X_7$ ~X_3~~

X stabilizers:

```
XIXIXIX
IXXIIXX
IIIXXXX
```

Smarter verification for Steane code

- Observe: X errors are correlated, but not arbitrary.



- Assume at most one X error occurs during preparation. What are the possible errors on the output?
  - Arbitrary single-bit errors (of course)
  - But what else?

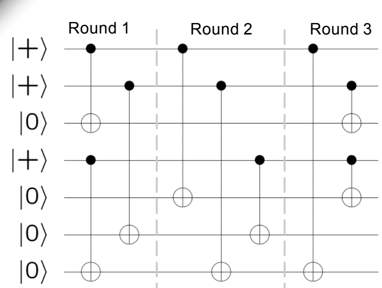
$\cancel{X_1 X_3 X_5 X_7 \quad \sim I}$

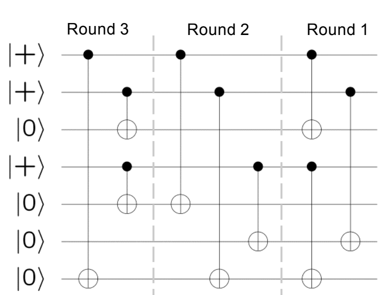
$\cancel{X_1 X_5 X_7 \quad \sim X_3}$

X_1X_7
X_2X_3
X_4X_5

X stabilizers:
```
XIXIXIX
IXXIIXX
IIIXXXX
```

Smarter verification for Steane code





- With one X error during preparation, what are the possible output errors?
  - Arbitrary single-bit errors, and

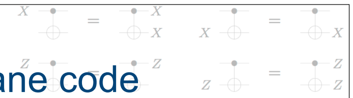
    X_1X_7
    X_2X_3 → correct!
    X_4X_5

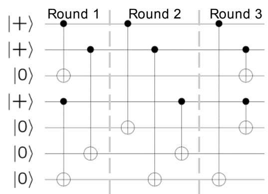
Conclusion: Applying CNOTs from a 123 ancilla into a 321 ancilla, correlated output errors from a single gate error can be distinguished, and *corrected* for.

  - Arbitrary single-bit errors, and

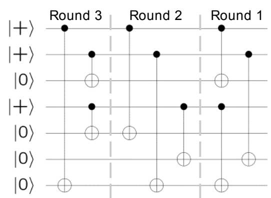
    X_1X_3
    X_2X_6 → don't correct!
    X_4X_7

Smarter verification for Steane code





- With one X error during preparation, possible output errors are:
  - Arbitrary single-bit errors, and
    X_1X_7
    X_2X_3 → correct!
    X_4X_5



  - Arbitrary single-bit errors, and
    X_1X_3
    X_2X_6 → don't correct!
    X_4X_7

Conclusion:
Applying CNOTs from a 123 ancilla into a 321 ancilla, correlated output errors from a single gate error can be distinguised, and *corrected* for. Postselection on no detected errors is not necessary. [Aliferis]

Consequences:
- No need for ancilla to wait for measurement results before using it.
- Reduced overhead.
- Provable threshold increases, but ancilla reliability may decrease.

Golay code preparation and verification

Stabilizers:

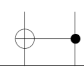
```
X.X..X..XXXXX..........
XXXX.XX.X....X.........
.XXXX.XX.X....X........
..XXXX.XX.X....X.......
...XXXX.XX.X....X......
X.X.X.XXX..X.....X.....
XXXX...X..XX......X....
XX.XXX...XX........X...
.XX.XXX...XX........X..
X..X..XXXXX..........X.
.X..X..XXXXX..........X
```

Golay code preparation and verification

Preparation circuit (shorthand):

```
1.2..3..4567X..........
2345.67.1....X.........
.2345.67.1....X........
..5671.23.4....X.......
...7143.56.2....X......
3.7.2.156..4.....X.....
4562...1..73......X....
51.367...42........X...
.71.452...36........X..
6..1..43725..........X.
.6..3..42715..........X
```

7 rounds |0⟩s |+⟩s

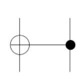


Golay code preparation and verification

Preparation circuit (shorthand):

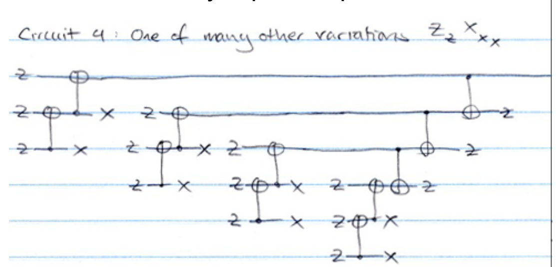
```
1.2..3..4567X..........
2345.67.1....X.........
.2345.67.1....X........
..5671.23.4....X.......
...7143.56.2....X......
3.7.2.156..4.....X.....
4562...1..73......X....
51.367...42........X...
.71.452...36........X..
6..1..43725..........X.
.6..3..42715..........X
```

|0⟩s |+⟩s



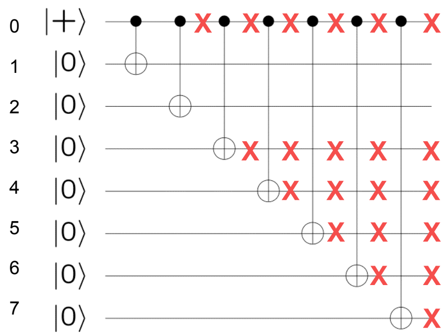
7 rounds

Verification by repeated postselection:



Golay code correlated errors

Abstract out:
XXXXXXXX



Possible output errors from single X failure:
Xs on

```
01234567 ~ ∅
0 234567 ~ 1
0  34567 ~ 12
0   4567 ~ 123
0    567 ~ 1234
0     67 ~ 12345
0      7 ~ 123456
```

If we reversed the rounds…

```
07654321 ~       ∅
0 654321 ~       7
0  54321 ~      67
0   4321 ~     567
0    321 ~    4567
0     21 ~   34567
0      1 ~  234567
```

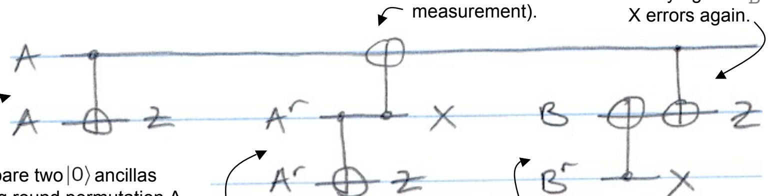
Possible output errors from two X failures:
consecutive sequences [a,b] = [a,a+1,…,b-1,b] e.g. 2345

Golay code final preparation and encoding circuits

Round permutations:
A=1243567
B=6274531
A^r=7653421
B^r=1354726

Verify once against Z errors (transverse X measurement).

Verify against X errors again.

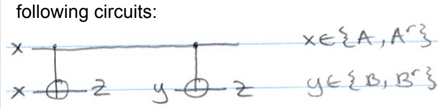


Prepare two |0⟩ ancillas using round permutation A. Verify once against X errors (measuring transversely in Z eigenbasis).

Prepare two |0⟩ ancillas using round permutation A^r (A reversed). Verify once against X errors.

Prepare two ancillas using round permutations B and B^r. Verify against Z errors.

Checking fault-tolerance reduces to checking following circuits:

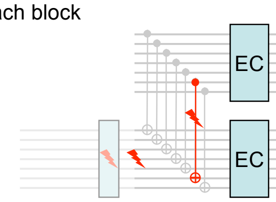


Conclusion:

- Reduces verification circuit complexity by half.
- Reduces overhead esp. at high error rates.
- Increases provable threshold (reduced combinatorial complexity allows much better computer-aided counting analysis).
- But ancilla reliability may decrease.

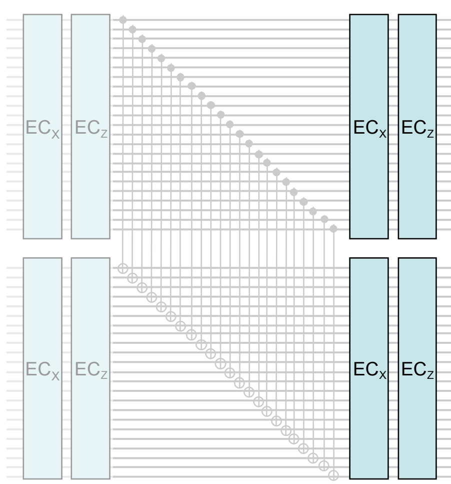
Analysis

- Aharonov & Ben-Or threshold proof:
  - Idea: Maintain inductive invariant of (1-)goodness. (A good block "has at most one bad subblock.")
  - Inefficient analysis:
    - $p \to \binom{m}{2} p^2$ not cp^3 for a distance-five code
    - No threshold for concatenated distance-three codes
- [R '05, Aliferis/Gottesman/Preskill '05] proofs apply too to distance-three codes
  - Idea: Maintain as inductive invariant recursive control over the probability distribution of errors in each block



  - Gives rigorous (and fairly efficient) criterion for threshold

Combinatorial analysis



Conclusion

- Technical background
  - Error correction
  - Stabilizer algebra
  - Quantum ECCs
- Ancilla preparation and verification
  - Steane preparation and heuristic verification
    - for Steane 7-qubit, distance-3 code
    - for Bacon/Shor 9-qubit, distance-3 code
    - for higher-distance codes
  - Strictly fault-tolerant verification
    - repeated purification
    - tweaked
- Rigorous noise threshold for 23-qubit, distance-7 Golay code
  - Technical setup
  - Combinatorial analysis

- Result: Threshold of 9.8×10^{-5}, or > 10^{-4} with 99.9% statistical confidence.
- Simulations haven't been run to estimate actual improvement.
- Other effects, particularly locality, still need to be analyzed.
- Analyze schemes which aren't strictly fault-tolerant.
- Consider schemes with no verification required.

- **Idea:** Differently prepare ancillas to verify against each other
  - No postselection for Steane code [Aliferis]
  - Halves preparation complexity for 23-qubit Golay code [Y. Ouyang, B.R.]